

MANUAL DE USUARIO

SOFTWARE QUE RECOPILA LOS DIFERENTES MÉTODOS DE **CRIPTOGRAFÍA**

Seguridad Informática

INTEGRANTES:

- ALCANTARA ZUÑIGA, Alex Rodolfo
- PANTA PISCOCHE, Jose Diego
- RAMOS ENCARNACION, Nilton
- TREJO OBREGON, Rodrigo Emilio
- VASQUEZ RAMOS, Jose Manuel



UNS
UNIVERSIDAD
NACIONAL DEL SANTA

UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas e Informática



SEGURIDAD INFORMÁTICA

IX CICLO – 2025 01

SOFTWARE QUE RECOPILA LOS DIFERENTES

MÉTODOS DE CRIPTOGRAFÍA

INTEGRANTES:

- ALCANTARA ZUÑIGA, Alex Rodolfo
- PANTA PISCOCHE, Jose Diego
- RAMOS ENCARNACION, Nilton
- TREJO OBREGON, Rodrigo Emilio
- VASQUEZ RAMOS, Jose Manuel

DOCENTE:

Dr. GIL ALBARRAN GUILLERMO EDWARD

NUEVO CHIMBOTE – PERÚ

Julio, 2025

ÍNDICE DE CONTENIDO

I.	INSTALACIÓN.....	4
1.1.	Requisitos Previos	4
1.2.	Clonar el Repositorio	4
1.3.	Crear un Entorno Virtual.....	4
1.4.	Activar el Entorno Virtual.....	5
1.5.	Instalar las Dependencias	5
1.6.	Ejecutar la Aplicación.....	5
II.	RESUMEN TEÓRICO DE LOS MÓDULOS DEL SISTEMA	7
2.1.	Sistemas de Cifra en Flujo	7
2.2.	Cifrado Simétrico en Bloque	7
2.3.	Cifrado Asimétrico con Mochilas	7
2.4.	Cifrado Asimétrico Exponencial.....	7
2.5.	Funciones Hash en Criptografía	7
2.6.	Autenticación y Firma Digital.....	8
2.7.	Certificados Digitales	8
2.8.	Criptografía Cuántica	8
2.9.	Blockchain	8
III.	DEMOSTRACIÓN DE LA APLICACIÓN	9
3.1.	Página de Inicio	9
3.2.	Cifrado en flujo	10
3.3.	Cifrado Simétrico en Bloque	13
3.4.	Cifrado Asimétrico con Mochilas	15
3.5.	Cifrado Asimétrico Exponencial.....	17
3.6.	Funciones Hash	20
3.7.	Autenticación y Firma Digital.....	22
3.8.	Certificados Digitales	25
3.9.	Criptografía Cuántica	29
3.10.	Blockchain	33
IV.	CONCLUSIONES.....	36

I. INSTALACIÓN

1.1. Requisitos Previos

- **Python:** Puedes verificar si tienes Python instalado ejecutando `python --version` en la terminal. Si no, puedes descargarlo desde <https://www.python.org/downloads/>
- **Git:** Para clonar el repositorio. Puedes verificar su instalación con `git --version`. Descárgalo desde <https://git-scm.com/downloads>
- **pip:** El gestor de paquetes de Python. Generalmente viene instalado con Python.

Una vez verificados los requisitos, sigue estos pasos detallados para ejecutar el proyecto en tu máquina local.

1.2. Clonar el Repositorio

Lo primero que debemos hacer es clonar el repositorio desde GitHub para obtener los archivos del proyecto en tu máquina local. Abre tu terminal o línea de comandos y ejecuta el siguiente comando:

```
git clone https://github.com/josevasquezramos/seguridadinformatica_criptografia.git
```

Este comando descargará una copia completa del proyecto en tu directorio actual.

Después de clonar el repositorio, ingresa al directorio del proyecto con el siguiente comando:

```
cd seguridadinformatica_criptografia
```

Esto cambiará el directorio de trabajo a la carpeta del proyecto.

1.3. Crear un Entorno Virtual

El siguiente paso es crear un entorno virtual, lo cual es una buena práctica en desarrollo de software, ya que te permite tener una instalación de Python aislada con las dependencias necesarias para este proyecto sin interferir con otras configuraciones en tu sistema.

Ejecuta el siguiente comando para crear un entorno virtual dentro del proyecto:

```
python -m venv venv
```

Este comando creará una carpeta llamada `venv` dentro de tu proyecto, donde se almacenarán todas las dependencias de Python específicas para este proyecto.

1.4. Activar el Entorno Virtual

Una vez creado el entorno virtual, necesitamos activarlo para poder usar las librerías de Python que se instalarán en él.

- **En Windows:**

Para activar el entorno virtual en Windows, utiliza el siguiente comando:

```
venv\Scripts\activate
```

Verás que el nombre del entorno virtual (venv) aparece al inicio de la línea de comandos, lo que indica que está activo.

- **En Linux/Mac:**

En sistemas basados en Linux o Mac, el comando para activar el entorno es:

```
source venv/bin/activate
```

Al igual que en Windows, verás que el nombre del entorno (venv) aparece en tu terminal, confirmando que el entorno virtual está activo.

1.5. Instalar las Dependencias

Una vez el entorno virtual esté activo, necesitas instalar las dependencias necesarias para que el proyecto funcione correctamente. Las dependencias están listadas en el archivo requirements.txt del proyecto.

Ejecuta el siguiente comando para instalar todas las dependencias:

```
pip install -r requirements.txt
```

Este comando descargará e instalará todas las bibliotecas necesarias como se especifica en el archivo requirements.txt.

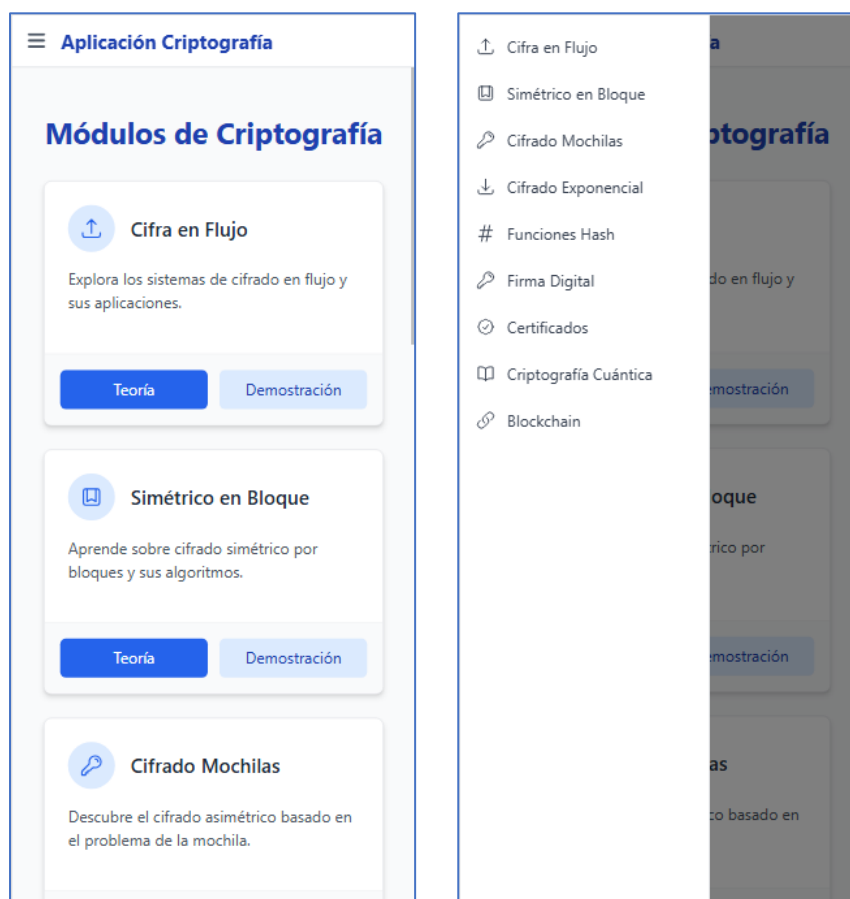
1.6. Ejecutar la Aplicación

Con todas las dependencias instaladas, ya estás listo para ejecutar la aplicación. En la terminal, escribe el siguiente comando para iniciar el servidor de la aplicación:

```
python criptografia/app.py
```

Esto ejecutará el archivo principal de la aplicación (app.py) ubicado en la carpeta criptografia, y podrás ver la aplicación corriendo localmente en tu máquina.

Una vez que la aplicación esté corriendo localmente, abre tu navegador web y accede al siguiente enlace: <http://127.0.0.1:8080/>



II. RESUMEN TEÓRICO DE LOS MÓDULOS DEL SISTEMA

2.1. Sistemas de Cifra en Flujo

Un sistema de cifra en flujo es un tipo de algoritmo criptográfico que cifra los datos en bits o en caracteres de forma continua, a medida que se reciben. A diferencia del cifrado en bloque, que cifra datos en bloques fijos, en los sistemas de cifra en flujo, se utiliza una clave generada de forma pseudo-aleatoria para transformar los datos, en general en tiempo real. Son especialmente útiles para aplicaciones donde los datos llegan de forma continua, como en la transmisión de voz o video.

2.2. Cifrado Simétrico en Bloque

El cifrado simétrico en bloque es un tipo de cifrado en el cual los datos son cifrados en bloques de longitud fija, utilizando la misma clave tanto para cifrar como para descifrar. Un ejemplo popular de cifrado simétrico en bloque es el AES (Advanced Encryption Standard). Este método es eficiente y seguro, pero la principal desventaja es que ambas partes deben compartir y proteger la misma clave secreta.

2.3. Cifrado Asimétrico con Mochilas

El cifrado asimétrico con mochilas es un sistema de cifrado basado en un problema matemático relacionado con las mochilas, en el cual se resuelve un conjunto de sumas o combinaciones. Cada usuario tiene un par de claves, una pública y una privada, y la seguridad del sistema se basa en la dificultad de resolver el problema de la mochila. Este tipo de cifrado se utiliza principalmente en sistemas de clave pública, como el algoritmo RSA.

2.4. Cifrado Asimétrico Exponencial

El cifrado asimétrico exponencial se refiere a sistemas de cifrado en los que las claves se basan en operaciones de exponentes y logaritmos. Un ejemplo común de este tipo de cifrado es el algoritmo RSA. En este sistema, las claves pública y privada son matemáticamente diferentes, pero relacionadas. La seguridad se basa en la dificultad de resolver el problema de la factorización de grandes números.

2.5. Funciones Hash en Criptografía

Las funciones hash son algoritmos que transforman una entrada de datos de longitud variable en una cadena de longitud fija, generalmente un número o una secuencia de caracteres. Estas funciones son unidireccionales (es decir, no se puede obtener la entrada original a partir del

hash) y se utilizan ampliamente en criptografía para verificar la integridad de los datos, generar firmas digitales y en la construcción de contraseñas seguras.

2.6. Autenticación y Firma Digital

La autenticación y la firma digital son procesos que garantizan la identidad y la integridad de un mensaje o documento. La autenticación asegura que un mensaje proviene de la persona que dice ser, y la firma digital es una forma de verificación criptográfica de la autenticidad y no alteración del mensaje. En este proceso, se utilizan claves privadas para firmar el mensaje y claves públicas para verificar la firma.

2.7. Certificados Digitales

Un certificado digital es un documento electrónico que autentica la identidad de una persona, organización o dispositivo en una red. Contiene información como el nombre del titular, la clave pública y la firma de una autoridad de certificación (CA). Su principal función es garantizar que las claves públicas en el certificado corresponden realmente a la persona o entidad que se dice ser, permitiendo un intercambio de información seguro.

2.8. Criptografía Cuántica

La criptografía cuántica es un campo emergente de la criptografía que utiliza principios de la mecánica cuántica para mejorar la seguridad de las comunicaciones. Se basa en el principio de superposición y entrelazamiento cuántico para proteger la información de interferencias y observaciones externas, haciendo que cualquier intento de espiar una transmisión cambie el estado de los datos, alertando de una posible amenaza. Un ejemplo conocido es el protocolo BB84 para el intercambio seguro de claves.

2.9. Blockchain

Blockchain es una tecnología de registro descentralizado que permite la creación de un libro de contabilidad digital, distribuido entre varios nodos, de manera que ninguna entidad central tiene control total sobre los registros. Los bloques de datos, que contienen transacciones, están vinculados entre sí mediante técnicas criptográficas, lo que hace que sean casi imposibles de alterar una vez añadidos. Blockchain se utiliza principalmente en criptomonedas, como Bitcoin, pero también tiene aplicaciones en áreas como contratos inteligentes y gestión de cadenas de suministro.

III. DEMOSTRACIÓN DE LA APLICACIÓN

3.1. Página de Inicio

Aquí podremos encontrar el índice de todos los módulos y links a las páginas de teorías y demostraciones de los algoritmos.

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📖 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 📄 Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Módulos de Criptografía



Cifra en Flujo

Explora los sistemas de cifrado en flujo y sus aplicaciones.

[Teoría](#)[Demostración](#)

Simétrico en Bloque

Aprende sobre cifrado simétrico por bloques y sus algoritmos.

[Teoría](#)[Demostración](#)

Cifrado Mochilas

Descubre el cifrado asimétrico basado en el problema de la mochila.

[Teoría](#)[Demostración](#)

Cifrado Exponencial

Estudia los fundamentos del cifrado exponencial asimétrico.



Funciones Hash

Conoce las funciones hash y su importancia en criptografía.



Firma Digital

Aprende sobre autenticación y firmas digitales.

3.2. Cifrado en flujo

Teoría:

Aplicación Criptografía

↑ Cifra en Flujo

Teoría

🔗 Demostración

📁 Simétrico en Bloque

🔗 Cifrado Mochilas

↓ Cifrado Exponencial

Funciones Hash

🔗 Firma Digital

🔗 Certificados

📁 Criptografía Cuántica

🔗 Blockchain

Cifrado en Flujo (Stream Cipher)

Método de cifrado simétrico bit a bit



¿Qué es el cifrado en flujo?

Un cifrado en flujo es un método de cifrado simétrico donde los bits de texto plano se combinan con un flujo de bits de clave cifrada (usualmente mediante la operación XOR).



Características principales

- Opera bit a bit o byte a byte
- Generalmente más rápido que el cifrado por bloques
- Requiere una clave tan larga como el mensaje
- Vulnerable si se reutiliza la clave



Algoritmo XOR

Nuestra implementación utiliza el operador XOR (OR exclusivo) para combinar el texto con la clave:

```
texto_cifrado = texto_plano XOR clave
```

Para descifrar se aplica la misma operación:

```
texto_plano = texto_cifrado XOR clave
```

Aplicación Criptografía

↑ Cifra en Flujo

Teoría

🔗 Demostración

📁 Simétrico en Bloque

🔗 Cifrado Mochilas

↓ Cifrado Exponencial

Funciones Hash

🔗 Firma Digital

🔗 Certificados

📁 Criptografía Cuántica

🔗 Blockchain

combinan con un flujo de bits de clave cifrada (usualmente mediante la operación XOR).

bloques

- Requiere una clave tan larga como el mensaje
- Vulnerable si se reutiliza la clave



Algoritmo XOR

Nuestra implementación utiliza el operador XOR (OR exclusivo) para combinar el texto con la clave:

```
texto_cifrado = texto_plano XOR clave
```

Para descifrar se aplica la misma operación:

```
texto_plano = texto_cifrado XOR clave
```

Prueba el cifrado en flujo

Experimenta con nuestro demostrador interactivo para entender cómo funciona este sistema de cifrado.

▶ [Ir a demostración interactiva](#)

Demostración:

Instrucciones

- Para cifrar: Ingresa texto plano y una clave, selecciona "Cifrar"
- Para descifrar: Ingresa texto en hexadecimal y la misma clave usada para cifrar, selecciona "Descifrar"
- La misma clave debe usarse para cifrar y descifrar
- No reutilices claves para diferentes mensajes

Aplicación Criptografía

Demostración de Cifrado en Flujo
Experimenta con el cifrado XOR interactivo

Texto
UNIVERSIDAD NACIONAL DEL SANTA

Clave
SISTEMAS

Modo
☒ Cifrar ☐ Descifrar

Procesar **Limpiar**

Aplicación Criptografía

Modo
☒ Cifrar ☐ Descifrar

Procesar **Limpiar**

Resultado
06071a02001f121a170817740b0c021a1c0712186509041f731a121a110c

Texto cifrado (hexadecimal)

Copiar

Aplicación Criptografía

↑ Cifra en Flujo

Teoría

Demostración

Simétrico en Bloque

Cifrado Mochilas

Cifrado Exponencial

Funciones Hash

Firma Digital

Certificados

Criptografía Cuántica

Blockchain

Demostración de Cifrado en Flujo

Experimenta con el cifrado XOR interactivo

Texto

06071a02001f121a170817740b0c021a1c0712186509041f731a121a110c

Clave

SISTEMAS

Modo

☐ Cifrar ☒ Descifrar

Procesar

Limpiar

Aplicación Criptografía

↑ Cifra en Flujo

Teoría

Demostración

Simétrico en Bloque

Cifrado Mochilas

Cifrado Exponencial

Funciones Hash

Firma Digital

Certificados

Criptografía Cuántica

Blockchain

Modo

☐ Cifrar ☒ Descifrar

Procesar

Limpiar

Resultado

UNIVERSIDAD NACIONAL DEL SANTA

Texto descifrado

Copiar

Instrucciones

- Para cifrar: Ingresa texto plano y una clave, selecciona "Cifrar"
- Para descifrar: Ingresa texto en hexadecimal y la misma clave usada para cifrar, selecciona "Descifrar"
- La misma clave debe usarse para cifrar y descifrar
- No reutilices claves para diferentes mensajes

3.3. Cifrado Simétrico en Bloque

Teoría:

Aplicación Criptografía

↑ Cifra en Flujo

📁 Simétrico en Bloque

📄 Teoría

🖼️ Demostración

👉 Cifrado Mochilas

↓ Cifrado Exponencial

Funciones Hash

👉 Firma Digital

🔒 Certificados

📖 Criptografía Cuántica

🔗 Blockchain

🔒 Cifrado Simétrico en Bloque (AES)

El estándar de cifrado avanzado más utilizado en la actualidad

📌 ¿Qué es el cifrado en bloque?

Un cifrado en bloque es un algoritmo de cifrado simétrico que opera sobre grupos de bits de longitud fija (bloques). **AES (Advanced Encryption Standard)** es el estándar actual más utilizado, aprobado por el NIST en 2001 como reemplazo de DES.

🛡️ Características principales

- ✓ Opera sobre bloques de **128 bits** (16 bytes)
- ✓ **10, 12 o 14 rondas** según tamaño de clave
- ✓ Claves de **128, 192 o 256 bits**
- ✓ Más seguro que los cifrados en flujo básicos

⚡ Prueba el cifrado AES

Aplicación Criptografía

↑ Cifra en Flujo

📁 Simétrico en Bloque

📄 Teoría

🖼️ Demostración

👉 Cifrado Mochilas

↓ Cifrado Exponencial

Funciones Hash

👉 Firma Digital

🔒 Certificados

📖 Criptografía Cuántica

🔗 Blockchain

⚡ Prueba el cifrado AES

Experimenta con nuestro demostrador interactivo para entender cómo funciona este sistema de cifrado estándar en la industria.

🔗 Ir a demostración interactiva

💡 Algoritmo AES (Modo CBC)

Nuestra implementación utiliza AES en modo CBC (Cipher Block Chaining), que añade un vector de inicialización para mayor seguridad:

```
Proceso de cifrado:
1. Padding PKCS#7 - Completa el texto para múltiplos de 16 bytes
2. Generación de IV - Vector de inicialización aleatorio (16 bytes)
3. Cifrado por bloques - Aplicación de transformaciones:
   - SubBytes
   - ShiftRows
   - MixColumns
   - AddRoundKey
4. Concatenación - IV + texto cifrado en Base64
```

Seguridad en CBC

El modo CBC protege contra patrones repetitivos al XOR cada bloque con el cifrado del bloque anterior, comenzando con el IV. Esto asegura que mensajes idénticos produzcan resultados diferentes.

Demostración:

Instrucciones

- Para cifrar: Ingresa texto plano y una clave de exactamente 32 caracteres
- Para descifrar: Ingresa texto en Base64 (incluye IV) y la misma clave usada para cifrar
- El sistema usa AES-256-CBC con padding PKCS7
- El IV (Vector de Inicialización) se genera automáticamente y se incluye en el resultado cifrado
- Guarda la clave de forma segura, es esencial para el descifrado

Aplicación Criptografía

Cifra en Flujo

Simétrico en Bloque

Teoría

Demostración

Cifrado Mochilas

Cifrado Exponencial

Funciones Hash

Firma Digital

Certificados

Criptografía Cuántica

Blockchain

Demostración de Cifrado AES-256-CBC

Experimenta con el estándar de cifrado avanzado en modo CBC (Cipher Block Chaining)

Texto:

EPISI

Clave (32 caracteres para AES-256):

%H6vKamTbk4jx1usofQ87KPFzVgB%jMp

32/32

La clave debe tener exactamente 32 caracteres (256 bits)

Modo:

☒ Cifrar ☐ Descifrar

Procesar

Generar Clave

Limpiar

✓

Resultado

z/RnT1in1u00KJri9Q1qavRY6m+w2Lz7z/SHn103oqg=

Texto cifrado (Base64 - incluye IV)

Copiar

Descargar

3.4. Cifrado Asimétrico con Mochilas

Teoría:

Aplicación Criptografía

- ↕ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- 📄 Teoría
- 👤 Demostración
- ⬇ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡 Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

📦 Cifrado Asimétrico con Mochilas (Merkle-Hellman)

Uno de los primeros algoritmos de **criptografía asimétrica**, basado en la dificultad de resolver el problema de la *mochila o subset sum problem*.

🕒 ¿En qué consiste?

La idea central es usar una secuencia de números **supercreciente** como clave privada y transformarla en una clave pública mediante operaciones modulares que ocultan la estructura.

🔒 Clave Privada

- ✓ Secuencia supercreciente
- ✓ Módulo m mayor que la suma total
- ✓ Multiplicador w coprimo con m

🔑 Clave Pública

Secuencia derivada de aplicar $(w \times \text{elemento}) \bmod m$ a cada elemento de la secuencia privada.

> Etapas del Algoritmo

- 1 **Generar secuencia supercreciente**
Cada número es mayor que la suma de todos los anteriores.

Aplicación Criptografía

- ↕ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- 📄 Teoría
- 👤 Demostración
- ⬇ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡 Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

> Etapas del Algoritmo

- 1 **Generar secuencia supercreciente**
Cada número es mayor que la suma de todos los anteriores.
- 2 **Seleccionar m y w**
 m mayor que la suma total, y w coprimo con m .
- 3 **Calcular clave pública**
Cada elemento es $(w \times \text{elemento}) \bmod m$.
- 4 **Cifrado**
El mensaje se convierte en bits y se multiplica por la clave pública.
- 5 **Descifrado**
Se aplica el inverso modular de w para recuperar la suma y resolver la mochila inversa.

🛡 Conceptos Clave

Secuencia supercreciente
Serie donde cada número es mayor que la suma de todos los anteriores.

⚠ ¿Por qué ya no se usa?

✗ **Vulnerabilidades descubiertas**
Existen algoritmos que rompen versiones simples del esquema.

📖 Valor educativo

Sigue siendo una base importante para entender

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📁 Simétrico en Bloque
- 🔑 Cifrado Mochilas
 - 📄 Teoría
 - 🎓 Demostración
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

5 Descifrado

Se aplica el inverso modular de w para recuperar la suma y resolver la mochila inversa.



Conceptos Clave

Secuencia supercreciente

Serie donde cada número es mayor que la suma de todos los anteriores.

Inverso modular

Número que cumple $(w \times w^{-1}) \bmod m = 1$.

Problema de la mochila

Dado un conjunto de pesos y una suma total, encontrar el subconjunto que la genera.



¿Por qué ya no se usa?



Vulnerabilidades descubiertas

Existen algoritmos que rompen versiones simples del esquema.



Valor educativo

Sigue siendo una base importante para entender criptografía basada en problemas NP-completos.

⚡ Ver Demostración Práctica

Demostración:

Ingresa un mensaje y de clic en Ejecutar operación.

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📁 Simétrico en Bloque
- 🔑 Cifrado Mochilas
 - 📄 Teoría
 - 🎓 Demostración
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain



Demostración Práctica de Cifrado de Mochilas

Experimenta con el algoritmo de Merkle-Hellman para cifrado asimétrico

Selecciona una acción:

Cifrar mensaje

Mensaje a cifrar:

Ingresa tu mensaje aquí

Ejecutar Operación



Resultado del Cifrado

Mensaje original

EPISI

Representación binaria

01000101010100000100100101001101001001

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🔑 Cifrado Mochilas
 - 📖 Teoría
 - 🎓 Demostración
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Mensaje a cifrar:

Ejecutar Operación

Resultado del Cifrado

Mensaje original

EPISI

Representación binaria

0100010101010000010010010101001101001001

Valores cifrados

[967, 206, 842, 687, 842]

Clave pública usada

[82, 123, 287, 83, 248, 373, 10, 471]

3.5. Cifrado Asimétrico Exponencial

Teoría:

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
 - 📖 Teoría
 - 🎓 Demostración
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

🔒 Cifrado Asimétrico Exponencial

Algoritmo de Diffie-Hellman

El **Cifrado Exponencial** es la base de muchos algoritmos de criptografía asimétrica. El más representativo es el **intercambio de claves Diffie-Hellman**, que permite a dos usuarios generar una clave secreta compartida en un canal inseguro, sin necesidad de transmitir dicha clave.

🔗 Fundamento Matemático

- Basado en la operación de **exponenciación modular**: $g^a \bmod p$
- Su seguridad radica en la dificultad del **Problema del Logaritmo Discreto**:

conocer $g, p, g^a \bmod p \rightarrow$ encontrar a es computacionalmente muy difícil.

⚙️ Funcionamiento del protocolo Diffie-Hellman

- Se acuerdan públicamente dos números: un **primo** p y un **generador** g .
- Cada parte elige su clave privada secreta: a (Alicia) y b (Bob).
- Cada uno calcula su clave pública:

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ↓ Cifrado Exponencial
 - 📄 Teoría
 - 👤 Demostración
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Funcionamiento del protocolo Diffie-Hellman

- Se acuerdan públicamente dos números: un **primo** p y un **generador** g .
- Cada parte elige su clave privada secreta: a (Alicia) y b (Bob).
- Cada uno calcula su clave pública:

$$A = g^a \bmod p$$

$$B = g^b \bmod p$$

- Se intercambian A y B .
- Cada parte calcula la clave compartida:

$$K = B^a \bmod p = A^b \bmod p$$

📌 Ambas partes obtienen la misma clave secreta, sin que esta se transmita en ningún momento.

Aplicaciones reales

🔒 **Protocolo TLS**
Seguridad en conexiones HTTPS

🔒 **VPNs**
Como OpenVPN

🔒 **SSH**
Intercambio de claves seguras

🔒 **Sesiones seguras**
En múltiples sistemas

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ↓ Cifrado Exponencial
 - 📄 Teoría
 - 👤 Demostración
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

🔒 **Protocolo TLS**
Seguridad en conexiones HTTPS

🔒 **VPNs**
Como OpenVPN

🔒 **SSH**
Intercambio de claves seguras

🔒 **Sesiones seguras**
En múltiples sistemas

Ejemplo didáctico

Datos públicos

$p = 23$

$g = 5$

Clave compartida

$K = 13$

Alice

$a = 6$

$$A = 5^6 \bmod 23 = 8$$

Bob

$b = 15$

$$B = 5^{15} \bmod 23 = 2$$

👉 Ir a demostración interactiva

Demostración:

Aplicación Criptografía

- ↑ Cifra en Flujo
- Simétrico en Bloque
- Cifrado Mochilas
- ↓ Cifrado Exponencial
 - Teoría
 - Demostración
- # Funciones Hash
- Firma Digital
- Certificados
- Criptografía Cuántica
- Blockchain

Demostración - Cifrado Asimétrico Exponencial

Diffie-Hellman + AES

Modo: Manual

p (número primo grande): g (generador):

Clave privada de A (a): Clave privada de B (b):

Mensaje a cifrar (opcional):

Ejecutar

Aplicación Criptografía

- ↑ Cifra en Flujo
- Simétrico en Bloque
- Cifrado Mochilas
- ↓ Cifrado Exponencial
 - Teoría
 - Demostración
- # Funciones Hash
- Firma Digital
- Certificados
- Criptografía Cuántica
- Blockchain

Resultados

p: A = $g^a \text{ mod } p$:

g: B = $g^b \text{ mod } p$:

a (A): Clave secreta:

b (B):

Mensaje Cifrado

Mensaje original:

Cifrado (base64):

Descifrado:

3.6. Funciones Hash

Teoría:

Aplicación Criptografía

- ↕ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ⬇️ Cifrado Exponencial
- # Funciones Hash
 - 📖 Teoría
 - 👤 Demostración
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

🔒 Funciones Hash en Criptografía

Las **funciones hash** son algoritmos matemáticos que transforman cualquier entrada en una **huella digital** de longitud fija. Son fundamentales en la seguridad informática por su capacidad para verificar integridad de datos sin revelar el contenido original.

📌 Características de una buena función hash

- ✓ **Determinista**
Siempre genera el mismo hash para la misma entrada.
- ✓ **Irreversible**
No se puede recuperar el texto original desde el hash.
- ✓ **Eficiente**
Computacionalmente rápida de calcular.
- ✓ **Sensible a cambios**
Pequeños cambios en la entrada generan hashes completamente distintos.
- ✓ **Resistente a colisiones**
Difícil encontrar dos entradas distintas con el mismo hash.

🔊 Aplicaciones de las funciones hash

Aplicación Criptografía

- ↕ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ⬇️ Cifrado Exponencial
- # Funciones Hash
 - 📖 Teoría
 - 👤 Demostración
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

🔊 Aplicaciones de las funciones hash

- ✓ Verificar la integridad de archivos y mensajes
- ✓ Seguridad en Blockchain
- ✓ Almacenar contraseñas de forma segura
- ✓ Detección de duplicados o datos alterados
- ✓ Firmas digitales y certificados
- ✓ Técnicas de autenticación y validación

📌 Algoritmos hash más comunes

ALGORITMO	LONGITUD	SEGURIDAD	USO COMÚN
MD5	128 bits	Baja (vulnerable)	Checksums y pruebas simples
SHA-1	160 bits	Media (obsoleto)	Certificados antiguos
SHA-256	256 bits	Alta	Blockchain, SSL/TLS, JWT
SHA-512	512 bits	Muy alta	Seguridad avanzada

Demostración:

Ingrese un mensaje y seleccione cualquiera de los métodos hash.

Aplicación Criptografía

- Cifra en Flujo
- Simétrico en Bloque
- Cifrado Mochilas
- Cifrado Exponencial
- # Funciones Hash**
 - Teoría
 - Demostración**
- Firma Digital
- Certificados
- Criptografía Cuántica
- Blockchain

Demostración Educativa de Funciones Hash

Explora cómo funcionan los algoritmos hash criptográficos paso a paso

Texto a procesar:

Selecciona el algoritmo:

MD5 (128 bits)

Calcular Hash

Resultado del Hash

Texto original:
HOLA UNS

Algoritmo:
MD5

Aplicación Criptografía

- Cifra en Flujo
- Simétrico en Bloque
- Cifrado Mochilas
- Cifrado Exponencial
- # Funciones Hash**
 - Teoría
 - Demostración**
- Firma Digital
- Certificados
- Criptografía Cuántica
- Blockchain

HOLA UNS

Algoritmo:
MD5

Hash final:

eee13dd0ceee26631aea46885bcbede9

Explicación del Algoritmo

MD5 genera un hash de 128 bits. Es rápido pero vulnerable a colisiones. Se usa aún en comprobaciones simples de integridad.

Proceso Educativo por Bloques

BLOQUE (HEX)	CONTENIDO	DIGEST PARCIAL
484f4c4120554e53	HOLA UNS	eee13dd0ceee26631aea46885bcbede9

3.7. Autenticación y Firma Digital

Teoría:

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ⬇ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
 - 📄 Teoría
 - 👤 Demostración
- 🛡 Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Autenticación y Firma Digital

Teoría y Conceptos Fundamentales

La autenticación y firma digital son componentes fundamentales de la seguridad informática moderna, permitiendo verificar la identidad de los participantes en una comunicación digital y asegurar la integridad de los documentos electrónicos.

Autenticación Digital

Concepto

Proceso de verificación de la identidad de un usuario, sistema o entidad en el entorno digital.

Métodos

Firma Digital

Definición

Mecanismo criptográfico que verifica la autenticidad e integridad de mensajes o documentos digitales.

Características

Certificados Digitales

¿Qué son?

Documentos electrónicos que vinculan una clave pública con una identidad, emitidos por una Autoridad Certificadora (CA).

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ⬇ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
 - 📄 Teoría
 - 👤 Demostración
- 🛡 Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Métodos

- Sabes** Contraseñas, PINs, preguntas de seguridad
- Tienes** Tokens físicos, tarjetas inteligentes
- Eres** Biometría (huellas, reconocimiento facial)
- MFA** Autenticación multifactor

Protocolos

- Kerberos
- OAuth
- OpenID
- SAML

Consideraciones Legales

documentos digitales.

Características

- Autenticidad**
Verifica la identidad del firmante
- Integridad**
Asegura que no hubo alteraciones
- No repudio**
El firmante no puede negar la autoría

Proceso

1. Generar hash del documento
2. Cifrar hash con clave privada
3. Adjuntar firma al documento
4. Verificar con clave pública
5. Comparar hashes

pública con una identidad, emitidos por una Autoridad Certificadora (CA).

Estructura X.509

- Versión del certificado
- Número de serie
- Algoritmo de firma
- Emisor (CA)
- Periodo de validez
- Sujeto (propietario)
- Clave pública del sujeto
- Firma digital de la CA

PKI

Infraestructura de Clave Pública que gestiona certificados:

- CA
- RA
- Repositorio
- CRL

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📄 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
 - 📄 Teoría
 - 🖥️ Demostración
- ✓ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Legales

Validez Legal

- Uso de certificados reconocidos
- Dispositivos seguros de firma
- Protección de claves privadas

Regulaciones

- Reglamento eIDAS (UE)
- Ley de Firmas Electrónicas (EEUU)
- Normativas locales

Algoritmos

- RSA
- DSA
- ECDSA
- EdDSA

Desafíos de Seguridad

- ⚠️ Gestión segura de claves privadas
- ⚠️ Revocación de certificados
- ⚠️ Protección contra phishing
- ⚠️ Actualización de algoritmos

Aplicaciones Prácticas

- 🏦 Banca online
- 🛒 e-Commerce
- 📄 Document legales
- ✉️ Email seguro
- 🔒 Control de acceso
- 🔗 Validación de software

Demostración:

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📄 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
 - 📄 Teoría
 - 🖥️ Demostración
- ✓ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Demostración Real de Firma Digital



Generación de Firma Digital

Documento a firmar:

Contenido de documento

Generar Firma Digital

Resultados de Firma:

Hash SHA-256 del documento:

I91ZVkB117NMChF7PQ6suskfCbP0aNk9AthZtxJUE=

Firma digital (Base64):

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
 - 📄 Teoría
 - 🔗 Demostración
- 📜 Certificados
- 🔍 Criptografía Cuántica
- 🔗 Blockchain

Firma digital (Base64):

NPSepUWI0r7h8sD0kRBeT0XBDFC0ngUApj33x3+HJjXq609D+W4oZqU2N0jYHu/7E2X07LpmDqPpWFSIdYVvUK1tAMkQeoTc0I

Clave pública PEM:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0uHgOw5ckYLYkApXb
BseZvxtY5uIFJTF/wdvmyS3GvUfflJCSoASvRp0IEMN1rIf179EDRue7/nEEY/oX
XwGpmKQ8QNqB6VZGdNTP17T9e1LzWeM2cLgVTZ64xdMrOKR8k1eDhs5oP+g1+yy1
pbeJZtkJxadVSr7yHPusD2wuHG4eA5L7H2T/xUoTO55nJ6cSi2beLb/40rnL3uFL
EPZ+xbbr8PjNcEGSDrdwzUfVRhhzjN1OPL4vh4LCPJpdgpT7WcvxoxcoHJpheXiR
1GdSiDVledF7wJ5mIdncnGugEKXwgkHh8q9UcpRy6aa8QQv9T1RWUXMsFpaCdQC4
zwIDAQAB
-----END PUBLIC KEY-----
```

Verificación



Verificación de Firma

Documento original:

Contenido de documnto

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
 - 📄 Teoría
 - 🔗 Demostración
- 📜 Certificados
- 🔍 Criptografía Cuántica
- 🔗 Blockchain



Verificación de Firma

Documento original:

Contenido de documnto

Firma a verificar (Base64):

NPSepUWI0r7h8sD0kRBeT0XBDFC0ngUApj33x3+HJjXq609D+W4oZqU2N0jYHu/7E2X07LpmDqPpWFSIdYVvUK
ItAMkQeoTc0l1BqKcz+Q4+4G2hH0mOqjlButtwNNjQpGxILWZrCtIJBqYUqfSU000Q3wecIIIEI9I53K6Dx/mMmLn
bf8BDWYyK1+Sod4rj3cT3hXRupYJeFWRKrnosdf1KmDdJ3LXS0vdsLwa9BniOmAJQ8UDwQS60KCKZKoYw/OWTb
nSVGfol7Dg+fy5s/L2BKE6t6+U9jx9LewRRwH2ShYejGxU3vglu3ETD1vei0DKjXh5luDfobbdjGocsQ==

Clave pública del firmante (PEM):

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0uHgOw5ckYLYkApXb
BseZvxtY5uIFJTF/wdvmyS3GvUfflJCSoASvRp0IEMN1rIf179EDRue7/nEEY/oX
XwGpmKQ8QNqB6VZGdNTP17T9e1LzWeM2cLgVTZ64xdMrOKR8kleDhs5oP+g1+yyi
pbeJZtkJxadVSr7yHPusD2wuHG4eA5L7H2T/xUoTO55nJ6cSi2beLb/40rnL3uFL
EPZ+xbbr8PjNcEGSDrdwzUfVRhhzjN1OPL4vh4LCPJpdgpT7WcvxoxcoHJpheXiR
1GdSiDVledF7wJ5mIdncnGugEKXwgkHh8q9UcpRy6aa8QQv9T1RWUXMsFpaCdQC4
zwIDAQAB
-----END PUBLIC KEY-----
```


Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ⬇ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
 - 📄 Teoría
 - 🔗 Demostración
- 📜 Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
mOuHgOw5ckYLYkkApxXb
BseZvxy5ulFJTF/wdvmyS3GvUffjCS0ASvRpOIE
MNIrifi79EDRue7/nEEY/oX
XwGPmKQ8QNqB6VZGdNTPi7T9e1LzWeM2clgVTZ64xdMrOKR8kleDhs5oP+g1+yyi
pbeJZtkxadVSr7yHPusD2wuHG4eASL7H2T/xUoTO55nJ6cSi2beLb/40mL3uFL
EPZ+xbbr8PjNcEGSDrdwzuFVRhhzjNIOPL4vh4LCPjpdgpT7WcvxoxcoHJpheXiR
1GdSIDVledF7wJ5mldncnGugEKXwgkHh8q9UcpRy6aa8QQv9T1RWUXMsFpaCdQC4
zwIDAQAB
-----END PUBLIC KEY-----
```

Verificar Firma

Resultado de Verificación:

✅ FIRMA VÁLIDA - El documento es auténtico e íntegro

Documento original:

Contenido de documnto

3.8. Certificados Digitales

Teoría:

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ⬇ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 📜 Certificados
 - 📄 Teoría
 - 🔗 Demostración
- 📖 Criptografía Cuántica
- 🔗 Blockchain

Certificados Digitales

Teoría y Conceptos Fundamentales

🔗 Ir a Demostración Práctica



¿Qué es un Certificado Digital?

Un certificado digital es un documento electrónico que vincula una clave pública con una identidad (persona, organización o dispositivo) y es emitido por una Autoridad Certificadora (CA) confiable.



Estructura X.509

- **Versión:** Formato X.509 (v1, v2, v3)
- **Número de serie:** Identificador único
- **Algoritmo de firma:** Usado por la CA
- **Emisor:** Autoridad Certificadora
- **Validez:** Periodo de vigencia
- **Clave pública:** Algoritmo y clave



Infraestructura de Clave Pública (PKI)

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 📄 Certificados
 - 📖 Teoría
 - 👤 Demostración
- 📖 Criptografía Cuántica
- 🔗 Blockchain



Infraestructura de Clave Pública (PKI)

Componentes Principales

- **CA:** Autoridad Certificadora
- **RA:** Autoridad de Registro
- **Repositorio:** Certificados válidos

Mecanismos de Revocación

- **CRL:** Lista de Certificados Revocados
- **OCSP:** Verificación en tiempo real



Tipos de Certificados

- SSL** Autenticación de servidores web
- Cliente** Para personas físicas/jurídicas
- Código** Firma de software
- Email** Firma y cifrado (S/MIME)



Ciclo de Vida

- 1 Generación de claves
- 2 Solicitud (CSR)
- 3 Validación de identidad
- 4 Emisión por la CA



Jerarquía de Confianza



Formatos Comunes

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 📄 Certificados
 - 📖 Teoría
 - 👤 Demostración
- 📖 Criptografía Cuántica
- 🔗 Blockchain



Tipos de Certificados

- SSL** Autenticación de servidores web
- Cliente** Para personas físicas/jurídicas
- Código** Firma de software
- Email** Firma y cifrado (S/MIME)



Ciclo de Vida

- 1 Generación de claves
- 2 Solicitud (CSR)
- 3 Validación de identidad
- 4 Emisión por la CA



Jerarquía de Confianza

- ➔ **Root CA**
Autoridad raíz (auto-firmada)
- ➔ **Intermediate CA**
Autoridades intermedias
- ➔ **End-entity**
Certificados de usuario final



Formatos Comunes

- | | |
|------------------|----------------|
| PEM (.pem, .crt) | DER (binario) |
| PKCS#7 (.p7b) | PKCS#12 (.p12) |

Demostración:

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 👉 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🖨️ Firma Digital
- 🔑 Certificados
 - 📖 Teoría
 - 👤 Demostración
- 🔒 Criptografía Cuántica
- 🔗 Blockchain

Demostración - Certificados Digitales

Generación y verificación de certificados X.509

+ Generación de Certificado

Nombre común (CN)

uns.edu.pe

Generar Certificado

Certificado Generado

Certificado (PEM)

```
-----BEGIN CERTIFICATE-----
MIIDSjCCAJKgAwIBAgIUcWcZt4jy9hcar3l0ffY12LOF6EcdQY:
BQAwYjELMAkGA1UEBhMCVHxEZARBgNVBAgMKmhbG1mb3JuaE:
DVNhb1B6cmFuY21zY28xEDA0BgNVBAoMB0NBIER1bW8xFDAS
bW8uY29tM4XD01I1D0c0HjABHzY1NV0XDTI1MTAxMDA0MzY1NV:
BhMCVHxEZARBgNVBAgMKmhbG1mb3JuaEExETAPBgNVBAoNCER:
EQYDQD0Ap1bnMuZmR1LnB1M1I1BjANBgkqhkiG9w0BAQFAAD:
AQEAuGKkHe+/83SHBrfAg1hlyJ0N9Crogm4/+CG5fW+0LQTRd24:
K6bVBR22/frs13j1msweUx3tQh6idInGhtzk/+2oRDFLCENi/t:
U1nOmIplrt6JvndgumMQ0Kvhg1+4Umn/PUD0E06EZmkCXW7YXqf:
XwIFNZ1tYea15K901Eo1Rogh7G7K1pJfOaTeqR+CpwhG18Zt29Yt:
-----END CERTIFICATE-----
```

🔑 Verificación de Certificado

Certificado a verificar (PEM)

```
-----BEGIN CERTIFICATE-----
MIIDSjCCAJKgAwIBAgIUcWcZt4jy9hcar3l0ffY12LOF6EcdQY:
BQAwYjELMAkGA1UEBhMCVHxEZARBgNVBAgMKmhbG1mb3JuaE:
DVNhb1B6cmFuY21zY28xEDA0BgNVBAoMB0NBIER1bW8xFDAS
bW8uY29tM4XD01I1D0c0HjABHzY1NV0XDTI1MTAxMDA0MzY1NV:
BhMCVHxEZARBgNVBAgMKmhbG1mb3JuaEExETAPBgNVBAoNCER:
EQYDQD0Ap1bnMuZmR1LnB1M1I1BjANBgkqhkiG9w0BAQFAAD:
AQEAuGKkHe+/83SHBrfAg1hlyJ0N9Crogm4/+CG5fW+0LQTRd24:
K6bVBR22/frs13j1msweUx3tQh6idInGhtzk/+2oRDFLCENi/t:
U1nOmIplrt6JvndgumMQ0Kvhg1+4Umn/PUD0E06EZmkCXW7YXqf:
XwIFNZ1tYea15K901Eo1Rogh7G7K1pJfOaTeqR+CpwhG18Zt29Yt:
-----END CERTIFICATE-----
```

Certificado de la CA (PEM)

```
-----BEGIN CERTIFICATE-----
MIIDZTCCAK2gAwIBAgIU038c1shWz54r02YIw1tpUwTrcIw:
DQY3KoZlhcNAQEL
BQAwYjELMAkGA1UEBhMCVHxEZARBgNVBAgMKmhbG1mb3JuaE:
DVNhb1B6cmFuY21zY28xEDA0BgNVBAoMB0NBIER1bW8xFDAS
bW8uY29tM4XD01I1D0c0HjABHzY1NV0XDTI1MTAxMDA0MzY1NV:
BhMCVHxEZARBgNVBAgMKmhbG1mb3JuaEExETAPBgNVBAoNCER:
EQYDQD0Ap1bnMuZmR1LnB1M1I1BjANBgkqhkiG9w0BAQFAAD:
AQEAuGKkHe+/83SHBrfAg1hlyJ0N9Crogm4/+CG5fW+0LQTRd24:
K6bVBR22/frs13j1msweUx3tQh6idInGhtzk/+2oRDFLCENi/t:
U1nOmIplrt6JvndgumMQ0Kvhg1+4Umn/PUD0E06EZmkCXW7YXqf:
XwIFNZ1tYea15K901Eo1Rogh7G7K1pJfOaTeqR+CpwhG18Zt29Yt:
-----END CERTIFICATE-----
```

✓ Verificar Certificado

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 👉 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🖨️ Firma Digital
- 🔑 Certificados
 - 📖 Teoría
 - 👤 Demostración
- 🔒 Criptografía Cuántica
- 🔗 Blockchain

Certificado Generado

Certificado (PEM)

```
-----BEGIN CERTIFICATE-----
MIIDSjCCAJKgAwIBAgIUcWcZt4jy9hcar3l0ffY12LOF6EcdQY:
BQAwYjELMAkGA1UEBhMCVHxEZARBgNVBAgMKmhbG1mb3JuaE:
DVNhb1B6cmFuY21zY28xEDA0BgNVBAoMB0NBIER1bW8xFDAS
bW8uY29tM4XD01I1D0c0HjABHzY1NV0XDTI1MTAxMDA0MzY1NV:
BhMCVHxEZARBgNVBAgMKmhbG1mb3JuaEExETAPBgNVBAoNCER:
EQYDQD0Ap1bnMuZmR1LnB1M1I1BjANBgkqhkiG9w0BAQFAAD:
AQEAuGKkHe+/83SHBrfAg1hlyJ0N9Crogm4/+CG5fW+0LQTRd24:
K6bVBR22/frs13j1msweUx3tQh6idInGhtzk/+2oRDFLCENi/t:
U1nOmIplrt6JvndgumMQ0Kvhg1+4Umn/PUD0E06EZmkCXW7YXqf:
XwIFNZ1tYea15K901Eo1Rogh7G7K1pJfOaTeqR+CpwhG18Zt29Yt:
tsDwh19sTCKWdms/6btxtgH+2fPp0JcF+063xypV0FA6B2ACU:
ToE2onHwCQhdpB5L1TK3rwUhwIDAQABoxAwDjANBgNVHRMBAf8:
S1b3DQEBChUAA1BAQCAjCkEqZ1tEGHrncsR98f85c41nqJ5g7r:
HbrKo+4Uo7/kHbP+A4543E72zary7+vr+R0Q29m/E+Z5fRUVbzT:
UfVpDZa1CRex9hwd+gUkX6fCz3so4ExaHeefyAPVQpVm/xQthp:
HTSNOPopqN+bh5r5A0V+bv60nTXL+RzSSlccmBswkLbCR65YKcnR:
4k/KWPK7Evx4d56ytYaVI184QdQj0jEVHY7zH/10oRRKy0KZ1:
30Teu6f7tZGhIO7zJGqb0Rd6Ls0+kgq1D+GjJqF5
-----END CERTIFICATE-----
```

Clave privada (PEM)

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQFAEAASCBKcwggSjAgEAAoIBAQC:
t8CDWE3InQ303GicbJ/4IZ39b7QtBNF3bhtX0MEdt7EQUEzErptUC:
azB5The1ChQ701caG30T/7ahEhUsLTE2L+0ggpKd5a8+vdnR5Kc6:
YxDQq+GCX7hRaf89R05gPoRmaQJdbtgherp/RdK4Z1VjE08FagU:
SjVG1ChsbsriwN85pN6pH4KnCEaXm3P1gP+3pqYpAQNmG2wPC:
G3G2Bb7Z8+q1okL5+07rFG+1U580dFYA3Z5Z0fKvzrGjNogTaj:
MrevBTAA3AgHBAAECggEA0Yw76cYGI07Vp9CZCsV4PX1bPgXQ/Ah:
h058FPBQCSiUs450741Z6mhQRXnk+2dF1ZBt6Cxo3L35N0085Yal:
rKtnqK2NGEYAY6SmaH849R4Y32vAVEH4LCQ4PtCu1RjJ97EMay2r:
-----END PRIVATE KEY-----
```

Certificado de la CA (PEM)

```
-----BEGIN CERTIFICATE-----
MIIDZTCCAK2gAwIBAgIU038c1shWz54r02YIw1tpUwTrcIw:
DQY3KoZlhcNAQEL
BQAwYjELMAkGA1UEBhMCVHxEZARBgNVBAgMKmhbG1mb3JuaE:
DVNhb1B6cmFuY21zY28xEDA0BgNVBAoMB0NBIER1bW8xFDAS
bW8uY29tM4XD01I1D0c0HjABHzY1NV0XDTI1MTAxMDA0MzY1NV:
BhMCVHxEZARBgNVBAgMKmhbG1mb3JuaEExETAPBgNVBAoNCER:
EQYDQD0Ap1bnMuZmR1LnB1M1I1BjANBgkqhkiG9w0BAQFAAD:
AQEAuGKkHe+/83SHBrfAg1hlyJ0N9Crogm4/+CG5fW+0LQTRd24:
K6bVBR22/frs13j1msweUx3tQh6idInGhtzk/+2oRDFLCENi/t:
U1nOmIplrt6JvndgumMQ0Kvhg1+4Umn/PUD0E06EZmkCXW7YXqf:
XwIFNZ1tYea15K901Eo1Rogh7G7K1pJfOaTeqR+CpwhG18Zt29Yt:
-----END CERTIFICATE-----
```

✓ Verificar Certificado

Aplicación Criptografía

↕ Cifra en Flujo

🔒 Simétrico en Bloque

🎒 Cifrado Mochilas

⬇️ Cifrado Exponencial

Funciones Hash

📝 Firma Digital

🔑 Certificados

📖 Teoría

🔗 Demostración

🔒 Criptografía Cuántica

🔗 Blockchain

```
4K/KWwK/7eVw4d5yYvA1J1W4QdQJyUvVHY7ZH/100KXyUKZ11
3DIEu6F7ZGhI07z7GQ80Rd6Ls0xkgq1D+GjQqF5
-----END CERTIFICATE-----
```

Clave privada (PEM)

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQFAASCBKcwggSjAgEAAoIBAQC4
t8CDWE3InQ30jG1CbJ/4IZ79b7QtBNF3bhX0MEt7EQUEzErptUk
az85ThE1ChQJ01caG30T/7ahEMUsITE2L+0gpkd5a8+vdnR5Kc6)
YxDQq+GCX7hRaf89R05gPoRmaQJdbtgherp/RdK4Z1VJE0BFagUJ
SjVGiCHsbsrWmN85pH6pH4KnCEaXm3P1gP+3pqYpAqMtmG2wPCT
G3G28b728+q1okL5+07rFg+1U58DoFYA35zZ0furKVzrGjNogTaj
MrevBTA3AgHBAECggEAOYw76cYGI67Vp99CZCs44PX1bPgXQ/Ah:
h058FPBQC5Iw4S0741Z6mhQRXnk+2dF1Z8t6Cxo3L3SNO085Yal
rKtnqK2NGEYAY6SmaH849R4Y3ZvAEH4LCQ4PtCuIRj397EjMay2r
ct+qH7FEBKfnTxzxEOYDjahfTzc8+7KVjZn8JmzFvN1uQdd2vel
j/wr79VBu+hanIZMqXrFkh7NY1mkXQ7U11frk1GwHPKQI06bze
w0fh7GdwVz456FIBCACLWzgg9uligJRV++vbioTSEHQK8Q00dkZ:
cG2NXF3U1QONBcRFUAI01b1se92C+8+eFP0ZGEU7+mSD1QIA/X9i
07065c1AHhr1bFhupB+Z6fQs+YDGI dXKAcfGEUPF7esIUxvNAmC
zpumhVHG91pq7pRX1rG296XswkBgQDVPXCBolH8ZBzhEznFAY1
OzC996+6fEUYnm5TEGDEogDvHQK4zNwjCzZKEqPES4gZtkJf
kQuYXwP+f5X138Ht5f1/QV3GoXHYkhDq1064YU1ZmOyTdudh+6Xc
YXj6qfOnQBK8GEm2a5rJqumhGF1h3p0BKmQUHUzyArLH6pRC
mapX033NH3c7zX7uR026eQuKT/+n14GIRoADomYkTV0nCTu0Da
9v09kqZ/P3PuYwCo6UoF3qntsQaqo60mNUKEGy8muUvMKFCf
x1yEgAfKa5ZPVY506X3nrH9UGugdQ/STPQ51Un39fGKwXaIzZgf
eVyxUfCY15r1KaIsR3dE0kg6K2f7bqG5xm1Zmr1Q174R+HUA16:
K2DQKts23uC+Knc1gz71pTfOjUj2wCukH+k9RMAoGADHSp0oa
rHx0Zoj4nt3NQKPAVRD0454004bqAJDtQL8Fj4LvK6mBpLahx
8/PnJvLnbi41CCKaFxCya/L/CU63Enm8n08E3Gc4m2QRuIAJfjG5f
tDPBNTUppovZfC4VklYITJQ=
-----END PRIVATE KEY-----
```

Certificado de la CA (PEM)

```
x1yEgAfKa5ZPVY506X3nrH9UGugdQ/STPQ51Un39fGKwXaIzZgf
eVyxUfCY15r1KaIsR3dE0kg6K2f7bqG5xm1Zmr1Q174R+HUA16:
K2DQKts23uC+Knc1gz71pTfOjUj2wCukH+k9RMAoGADHSp0oa
rHx0Zoj4nt3NQKPAVRD0454004bqAJDtQL8Fj4LvK6mBpLahx
8/PnJvLnbi41CCKaFxCya/L/CU63Enm8n08E3Gc4m2QRuIAJfjG5f
tDPBNTUppovZfC4VklYITJQ=
-----END PRIVATE KEY-----
```

Certificado de la CA (PEM)

```
-----BEGIN CERTIFICATE-----
MIIDOTCCak2gAwIBAgIU38c1shWz54r02YIw1tpUwTrcIwDQY:
BQAwYjElMAkGA1UEBhMCVHhEzARBgNVBAgMCKhhbG1mb3JuaHE
DWhb1B6cmFuY2l2Y28xEOA0BgNVBAoMB0NBIERI1bX8xZDASBgN
bW8uY29tMB4XDTI1MDcxMjA0MDUwNFoXDTI1MDcxMjA0MDUwNFo
BHMCMVHhEzARBgNVBAgMCKhhbG1mb3JuaHEfJAUBGVBACMDVNf
Y28xEOA0BgNVBAoMB0NBIERI1bW8xZDASBgNVBAWIMC2NhLmRl
BgkqhkiG9w0BAQFAAOCAQ8AMIIBgKCAQAw21L7Rp5SuzC11Z:
kZJD3axccGUFkY1huceH0yC11RuZqDCNtImvM3TLTQ48dVuf4h02
aT3CDn04edCosbp1gfFRjNm4AwOR+LVb6CK2yJD43rmd1odelJt
Uv0QVrdAwOpG775HzL70pfZxgSFaWqE2AQ+NQwBFA894xHURLE
RA1bv3En207usukK28wFuz0p11kHFY1agXwTqyAw8RqJf85awJ
C/TTt0D6KCrAYpVLPe5n1t8jL65m31W05RRc2JdgoJAAM7YsK9
oxHwETAPBgNHRMBAf8EBTADAQH/HA0GCSqGSIb3DQEBCwUAA4IE
rZndydzcrUg4jF37FcmVr827yh7oIH1N/tv41JffqE15jTrb19
JUQXbAGGA97u9mNbK515n1Hh378tGJAXDn4zsasZut12pmIsxdl
iPXUEytU2LdR1ZgggFY60Lewy7FKwKACUQgJHCnzq2K1Iw+sH1f
GsJYbRQLiz6hq6QyAHSkx9kdhiaqEY1bh1nYK7VhDargFEKCK8/
11b3oxUQ/wuKXT3z+mh0zHPcdFdtRUS1RpACqLhZaImFP1PR7HF
HKDxHpAybtJe
-----END CERTIFICATE-----
```

Aplicación Criptografía

↕ Cifra en Flujo

🔒 Simétrico en Bloque

🎒 Cifrado Mochilas

⬇️ Cifrado Exponencial

Funciones Hash

📝 Firma Digital

🔑 Certificados

📖 Teoría

🔗 Demostración

🔒 Criptografía Cuántica

🔗 Blockchain

3.9. Criptografía Cuántica

Teoría:

Aplicación Criptografía

- ↕ Cífra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cífrado Mochilas
- ⬇ Cífrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡 Certificados
- 📖 **Criptografía Cuántica**
 - 📄 Teoría
 - 🖥 Demostración
- 🔗 Blockchain

⚡ Criptografía Cuántica: Conceptos y Teorías

Explorando la frontera de la seguridad informática mediante principios de mecánica cuántica

📌 Introducción a la Criptografía Cuántica

La criptografía cuántica representa un cambio de paradigma en seguridad informática, utilizando principios fundamentales de la mecánica cuántica para garantizar comunicaciones inviolables. A diferencia de los sistemas clásicos que dependen de la complejidad computacional, esta aproximación se fundamenta en leyes físicas inmutables. Su desarrollo responde a la amenaza que representan las computadoras cuánticas para los sistemas criptográficos tradicionales, ofreciendo seguridad incondicional basada en las propiedades cuánticas de la materia.

💡 Principios Fundamentales

Principio de Incertidumbre

Heisenberg demostró que ciertas

Entrelazamiento Cuántico

Partículas entrelazadas exhiben

Teorema de No Clonación

Es imposible crear una copia perfecta

Aplicación Criptografía

- ↕ Cífra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cífrado Mochilas
- ⬇ Cífrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡 Certificados
- 📖 **Criptografía Cuántica**
 - 📄 Teoría
 - 🖥 Demostración
- 🔗 Blockchain

💡 Principios Fundamentales

Principio de Incertidumbre

Heisenberg demostró que ciertas propiedades físicas de partículas cuánticas no pueden medirse simultáneamente con precisión absoluta. En criptografía, esto protege contra la interceptación: cualquier medición altera irrevocablemente el estado cuántico, revelando la intrusión.

Entrelazamiento Cuántico

Partículas entrelazadas exhiben correlaciones instantáneas independientes de la distancia que las separa. Esta propiedad permite protocolos como E91, donde la medición de una partícula determina inmediatamente el estado de su pareja entrelazada.

Teorema de No Clonación

Es imposible crear una copia perfecta de un estado cuántico desconocido sin alterar el original. Garantiza que un atacante no pueda duplicar información cuántica para análisis posterior sin ser detectado.

🛡 Protocolos Principales

BB84 (1984)

Desarrollado por Charles Bennett y Gilles Brassard, fue el primer protocolo práctico de distribución cuántica de claves.

- ✓ Utiliza estados de polarización de fotones en dos bases diferentes (rectilínea y diagonal)

E91 (1991)

Protocolo diseñado por Artur Ekert que utiliza pares de fotones entrelazados para establecer claves secretas.

- ✓ Aprovecha las correlaciones cuánticas entre partículas entrelazadas

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📁 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡️ Certificados
- 📖 **Criptografía Cuántica**
 - 📄 Teoría
 - 👤 Demostración
- 🔗 Blockchain

BB84 (1984)

Desarrollado por Charles Bennett y Gilles Brassard, fue el primer protocolo práctico de distribución cuántica de claves.

- ✓ Utiliza estados de polarización de fotones en dos bases diferentes (rectilínea y diagonal)
- ✓ La seguridad deriva del principio de incertidumbre: medir en la base incorrecta altera el estado

E91 (1991)

Protocolo diseñado por Artur Ekert que utiliza pares de fotones entrelazados para establecer claves secretas.

- ✓ Aprovecha las correlaciones cuánticas entre partículas entrelazadas
- ✓ La violación de las desigualdades de Bell detecta cualquier intento de interceptación

Teorías Relacionadas

Teoría de la Información Cuántica

Extiende los conceptos clásicos de bits a qubits, introduciendo superposición y entrelazamiento como recursos computacionales.

Computación Cuántica

Utiliza fenómenos cuánticos para realizar operaciones que superan exponencialmente las capacidades de las computadoras tradicionales.

Teleportación Cuántica

Protocolo que transfiere estados cuánticos entre ubicaciones distantes usando entrelazamiento y comunicación clásica.

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📁 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡️ Certificados
- 📖 **Criptografía Cuántica**
 - 📄 Teoría
 - 👤 Demostración
- 🔗 Blockchain

Teorías Relacionadas

Teoría de la Información Cuántica

Extiende los conceptos clásicos de bits a qubits, introduciendo superposición y entrelazamiento como recursos computacionales.

Computación Cuántica

Utiliza fenómenos cuánticos para realizar operaciones que superan exponencialmente las capacidades de las computadoras tradicionales.

Teleportación Cuántica

Protocolo que transfiere estados cuánticos entre ubicaciones distantes usando entrelazamiento y comunicación clásica.

Aplicaciones Prácticas

- ✓ **Distribución Cuántica de Claves (QKD)**
Sistemas comerciales ya disponibles para transmisión segura de claves criptográficas
- ✓ **Redes de Comunicación Seguras**
Implementaciones en redes bancarias, gubernamentales y militares
- ✓ **Protección Post-Cuántica**
Defensa contra futuros ataques con computadoras cuánticas

Desafíos Actuales

- ⚠️ **Limitaciones de Distancia**
La atenuación en fibras ópticas limita la transmisión a ~100-200 km sin repetidores
- ⚠️ **Repeaters Cuánticos**
La tecnología para repetidores cuánticos fiables aún está en desarrollo
- ⚠️ **Implementación Práctica**
Vulnerabilidades en dispositivos reales pueden explotarse en ataques tipo "side-channel"

Demostración:

Aplicación Criptografía

- ↑ Cifra en Flujo
- Simétrico en Bloque
- 🔒 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 📜 Certificados
- 🔐 Criptografía Cuántica**
 - 📖 Teoría
 - 👤 Demostración**
- 🔗 Blockchain

⚡ Demostración Interactiva - Protocolo BB84

Simulación del primer protocolo de distribución cuántica de claves usando polarización de fotones

🕒 Iniciar Demostración

➡ Paso a Paso

🔄 Reiniciar

👤 Alice (Emisor)

Bits Aleatorios

0 0 0
1 1 1
1 1

Bases de Polarización

+ + x
x x x
x +

Fotones Polarizados

🔗 Canal Cuántico

👤 Bob (Receptor)

Bases de Medición

+ + +
x x x
+ +

Resultados de Medición

0 0 1
1 1 1
1 1

👤 Alice (Emisor)

Bits Aleatorios

0 0 0
1 1 1
1 1

Bases de Polarización

+ + x
x x x
x +

Fotones Polarizados

1 2 3 4 5 6 7 8

🔗 Canal Cuántico

👤 Bob (Receptor)

Bases de Medición

+ + +
x x x
+ +

Resultados de Medición

0 0 1
1 1 1
1 1

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📁 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡️ Certificados
- 📖 **Criptografía Cuántica**
 - 📄 Teoría
 - 👤 Demostración
- 🔗 Blockchain



Reconciliación de Claves

Bases que Coinciden

Posición 1 Posición 2 Posición 4 Posición 5
Posición 6 Posición 8

Clave Secreta Compartida

0 0 1 1 1 1



¿Cómo funciona el protocolo BB84?

- 1 Alice genera bits aleatorios y elige bases de polarización**
Para cada bit, Alice elige aleatoriamente entre la base rectilínea (+) o diagonal (x)
- 2 Alice envía fotones polarizados a Bob**
Cada fotón se polariza según el bit y base elegidos por Alice
- 3 Bob mide los fotones en bases aleatorias**
Bob no conoce las bases usadas por Alice y debe adivinarlas
- 4 Alice y Bob comparan públicamente las bases usadas**
Descartan los bits donde usaron bases diferentes

Aplicación Criptografía

- ↑ Cifra en Flujo
- 📁 Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔗 Firma Digital
- 🛡️ Certificados
- 📖 **Criptografía Cuántica**
 - 📄 Teoría
 - 👤 Demostración
- 🔗 Blockchain



¿Cómo funciona el protocolo BB84?

- 1 Alice genera bits aleatorios y elige bases de polarización**
Para cada bit, Alice elige aleatoriamente entre la base rectilínea (+) o diagonal (x)
- 2 Alice envía fotones polarizados a Bob**
Cada fotón se polariza según el bit y base elegidos por Alice
- 3 Bob mide los fotones en bases aleatorias**
Bob no conoce las bases usadas por Alice y debe adivinarlas
- 4 Alice y Bob comparan públicamente las bases usadas**
Descartan los bits donde usaron bases diferentes
- 5 Verificación de seguridad**
Comparan públicamente un subconjunto de bits para detectar intrusos
- 6 Obtención de la clave secreta**
Los bits restantes forman la clave secreta compartida

3.10. Blockchain

Teoría:

Aplicación Criptografía

- ↑ Cifra en Flujo
- Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain
- 📄 Teoría
- 👤 Demostración



Blockchain: Fundamentos Criptográficos y Seguridad

La tecnología blockchain representa un paradigma revolucionario en criptografía y seguridad informática, ofreciendo un sistema descentralizado, inmutable y transparente.



Conceptos Fundamentales

1. Cadena de Bloques

Estructura de datos donde cada bloque contiene:

- ✓ Transacciones validadas
- ✓ Timestamp (marca de tiempo)
- ✓ Hash del bloque anterior
- ✓ Nonce (número usado en minería)

2. Descentralización

La red se mantiene por nodos distribuidos que:

- ✓ Verifican transacciones
- ✓ Almacenan copias idénticas
- ✓ Eliminan puntos únicos de fallo

3. Consenso Distribuido

Mecanismos que permiten acordar el estado válido:

- ✓ Proof-of-Work (PoW)
- ✓ Proof-of-Stake (PoS)
- ✓ Sin necesidad de confianza mutua

Aplicación Criptografía

- ↑ Cifra en Flujo
- Simétrico en Bloque
- 🔑 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain
- 📄 Teoría
- 👤 Demostración



Criptografía en Blockchain

Funciones Hash Criptográficas

Algoritmos como SHA-256 (Bitcoin) que:

- ✓ Producen outputs únicos
- ✓ Son determinísticos
- ✓ Resistentes a colisiones
- ✓ Protegen la integridad

Criptografía Asimétrica

Uso de pares de claves públicas/privadas para:

- ✓ Firmas digitales (ECDSA)
- ✓ Autenticación de transacciones
- ✓ Generación de direcciones

Merkle Trees

Estructuras de datos que permiten:

- ✓ Verificación eficiente
- ✓ Pruebas de inclusión
- ✓ Optimización del almacenamiento



Aspectos de Seguridad

Inmutabilidad

Los bloques no pueden alterarse gracias a:

- ✓ Funciones hash criptográficas

Transparencia y Pseudonimato

- ✓ Transacciones públicas

- ✓ Direcciones criptográficas

Modelos de Amenazas

Principales vectores de ataque:

- ⚠️ Ataques del 51%

Aplicación Criptografía

- ↕ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ⬇️ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain
 - 📄 Teoría
 - 🎭 Demostración



Aspectos de Seguridad

Inmutabilidad

Los bloques no pueden alterarse gracias a:

- ✓ Encadenamiento criptográfico
- ✓ Consenso distribuido
- ✓ Resistencia al 51% attack

Transparencia y Pseudonimato

- ✓ Transacciones públicas
- ✓ Direcciones criptográficas
- ✓ Sin identidades reales

Modelos de Amenazas

Principales vectores de ataque:

- ⚠️ Ataques del 51%
- ⚠️ Sybil attacks
- ⚠️ Vulnerabilidades en smart contracts



Teorías Avanzadas

Teoría de Juegos

Análisis de incentivos económicos para garantizar comportamiento honesto mediante teoría de juegos no cooperativos.

Zero-Knowledge Proofs

Verificar transacciones sin revelar información (ej: zk-SNARKs en Zcash).

Byzantine Fault Tolerance

Alcanzar consenso con nodos maliciosos (ej: PBFT en Hyperledger).

Demostración:

Aplicación Criptografía

- ↕ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 🎒 Cifrado Mochilas
- ⬇️ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain
 - 📄 Teoría
 - 🎭 Demostración



Demostración Interactiva de Blockchain

Experimenta con los conceptos fundamentales de la tecnología blockchain



Crear Transacción

Añadir Transacción



Minar Bloque

Convierte las transacciones pendientes en un nuevo bloque de la cadena.

Minar Bloque



Transacciones Pendientes

REMITENTE	DESTINATARIO	CANTIDAD
-----------	--------------	----------

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 👉 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain
 - 📄 Teoría
 - 🎭 Demostración

Transacciones Pendientes

REMITENTE	DESTINATARIO	CANTIDAD
Guillermo	Osorio	5

Cadena de Bloques

Bloque #0

Hash:
genesis_hash

Hash Anterior:
0

Transacciones (0):
Bloque génesis (sin transacciones)

Bloque #1

Hash:
simulated_hash_1

Hash Anterior:
genesis_hash

Transacciones (2):

Aplicación Criptografía

- ↑ Cifra en Flujo
- 🔒 Simétrico en Bloque
- 👉 Cifrado Mochilas
- ↓ Cifrado Exponencial
- # Funciones Hash
- 🔑 Firma Digital
- 🛡️ Certificados
- 📖 Criptografía Cuántica
- 🔗 Blockchain
 - 📄 Teoría
 - 🎭 Demostración

Bloque #1

Hash:
simulated_hash_1

Hash Anterior:
genesis_hash

Transacciones (2):

- ✓ Bob → Alice: 2
- ✓ Guillermo → Osorio: 5

¿Cómo funciona esta demostración?

- Transacciones pendientes**
Las transacciones se agregan al pool de transacciones pendientes antes de ser incluidas en un bloque.
- Minería de bloques**
Al minar, las transacciones pendientes se incluyen en un nuevo bloque que se añade a la cadena.
- Encadenamiento criptográfico**
Cada bloque referencia al anterior mediante su hash, creando una cadena inmutable.
- Inmutabilidad**
Una vez añadido, un bloque no puede alterarse sin invalidar todos los bloques posteriores.

IV. CONCLUSIONES

El desarrollo del software que recopila diferentes métodos de criptografía permitió consolidar un entendimiento profundo de los principios teóricos y prácticos de las técnicas de seguridad informática abordadas en el proyecto. A través de la implementación de módulos como cifrado en flujo, cifrado simétrico en bloque, cifrado asimétrico (mochilas y exponencial), funciones hash, autenticación y firma digital, certificados digitales, criptografía cuántica y blockchain, se logró demostrar la importancia y aplicabilidad de estas herramientas en la protección de la información.

Entre los hallazgos más relevantes se destaca:

- **Eficiencia y versatilidad:** Los algoritmos de cifrado simétrico (como AES) resultaron óptimos para proteger datos en tiempo real, mientras que los métodos asimétricos (como RSA) demostraron su utilidad en el intercambio seguro de claves.
- **Integridad y autenticidad:** Las funciones hash y las firmas digitales son fundamentales para garantizar la autenticidad de los mensajes y la detección de alteraciones.
- **Futuro de la criptografía:** Tecnologías emergentes como la criptografía cuántica y blockchain presentan un potencial disruptivo para resolver problemas de seguridad tradicionales, aunque su adopción aún enfrenta desafíos técnicos.